

# TRACK 7: PERFORMANCE SESSION 760

## ERD/TD REVIEWS FOR OPTIMUM DB2 PERFORMANCE

**RALPH NICOSIA  
NIMS ASSOCIATES, INC.**

### **Part One: ERD Review Objectives**

- ✓ *Identify constructs which may cause physical implementation issues*
- ✓ *Determine if issues can be addressed later*
- ✓ *Address issues that cannot be resolved in action diagrams or technical design*
- ✓ *Document issues to be addressed later*
- ✓ *Problems discovered here may cause a change to the ERD*
- ✓ *How can the business rules be maintained?*

## **Part One: ERD Review Objectives**

From: *A Guide To Information Engineering Using The IEF*

*"The analyst must ignore implementation details in order to accurately model business data. Later, during Technical Design stage, the designer may take steps to optimize the physical implementation derived from the ER model"*

Copyright 1996, Nims Associates, Inc.



## **Who Gets Invited?**

- DA, DBA
- Business Analyst
- IEF Technical Analyst
- Development Staff (project lead, lead analyst, others)
- IS and Client Management (formal session)



Copyright 1996, Nims Associates, Inc.



## ***What are the Review Materials?***

- ER Model
- Entity/Attribute Report
- Relationship Report
- Consistency Check output
- Public Interface access
  - Public Interface loaded with current model
  - Review point outputs (if automated)
- Issue documentation
- Data Structure Report
- Data Store Report
- RI Report
- Public Interface access
- Shop DBMS standards
- DB2 Catalog Access
- EXPLAIN analysis report
- ***NOTE: Items evolve from key-based issues to fully-attributed issues***

Copyright 1996, Nims Associates, Inc.



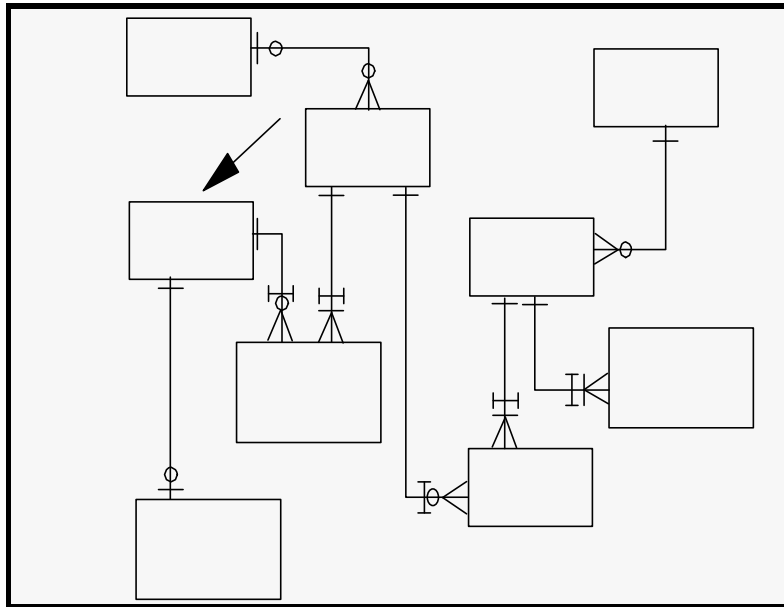
## ***Weighting of issues found***

- Important for "ranking" where time is spent:  
FIXES vs. FUTURE ENHANCEMENTS
- Usual items to rate:
  - Ease of detection
  - Ease of fix
  - Level of impact
- Determined by factors such as:
  - Upstream and Downstream impact
  - Expected performance improvement
  - Volume of re-occurring items

Copyright 1996, Nims Associates, Inc.



## *Example Model*

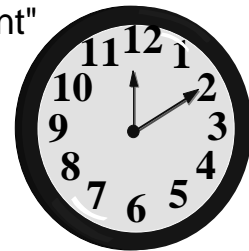


Copyright 1996, Nims Associates, Inc.



## *ERD Review Timing*

- During development or Post-development
- This presentation assumes "During Development"
  - Follows business review
  - Separate and different
  - Working sessions
    - Periodic, on average 3 times during development
    - Once with key-based model, others with fully attributed model
  - Formal session
    - When model is "*complete*"!!



Copyright 1996, Nims Associates, Inc.



## **Many-To-Many Relationships**

- "Associative" table not addressable by action diagram
- Hidden access in action diagram to "associative" table
- Intersection data, discovered later, cannot be implemented
- Confusing to end-users with ad-hoc access
- Review for:
  - Many-to-many relationship
- Suggested Action:
  - Implement associative entity in ER model

Copyright 1996, Nims Associates, Inc.



## **Many-To-Many Relationships**

- "One case where an entity type with no attributes should not be eliminated is when an associative entity type resolves a many-to-many relationship"
  - A Guide to Information Engineering Using The IEF



Copyright 1996, Nims Associates, Inc.



## **Many-To-Many Relationships**

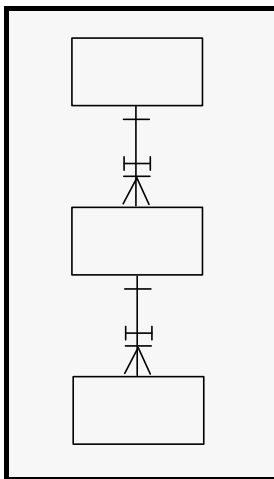
"Link Table"

Record	+-SUPPORTED_BY		12
FK Field	FK_EVENTNUMBER	Small	2
FK Field	FK_SPONSORNUM	Small	2
Linkage	<No Name> SPONSOR		
Linkage	<No Name> EVENT		
EntryPoint (U)	+-I0000519		4
Field	FK_SPONSORNUM	Small	2
Field	FK_EVENTNUMBER	Small	2
	+-		
EntryPoint (U)	+-I0000516 (Primary)		4
Field	FK_EVENTNUMBER	Small	2
Field	FK_SPONSORNUM	Small	2

Copyright 1996, Nims Associates, Inc.



## **Fully Mandatory at Bottom of Hierarchy**



- RI rules above a fully-mandatory construct cannot be DB2 supported
  - An "eye-opener" at this point of life-cycle
  - Can be an issue if DB2 RI is important
- Review for:
  - Fully mandatory relationship as a dependent in a cascade structure
- Suggested action:
  - If DB2 RI is critical, change relationship to optional-dependent

Copyright 1996, Nims Associates, Inc.



## **Modifying Relationships**

- IEF will force a read to the STATE code entity, prior to adding an APPLICANT
- Does the STATE code table need "locked" up while adding APPLICANTS?



Copyright 1996, Nims Associates, Inc.



## **Modifying Relationships**

- Also addressed at later phases
- Modifying relationships insure integrity of the children
  - But, can lead to contention problems
- Referencing relationships allow intervening DELETES of parents while ASSOCIATEing children
  - Can reduce contention issues

Copyright 1996, Nims Associates, Inc.



## **Modifying Relationships**

- Review for:
  - Parent in a modifying relationship with a large number of immediate dependents
  - Modifying relationship where parent is never to be DELETED\*
  - Parent/dependent relationship in which dependent has a high level of CREATE activity\*
- \* Reviewed with Action Blocks and Physical Data Design
- Suggested action:
  - Define relationship as REFERENCING if concurrency needs will outweigh integrity risks

Copyright 1996, Nims Associates, Inc.



## **Missing RI Rule Specification**

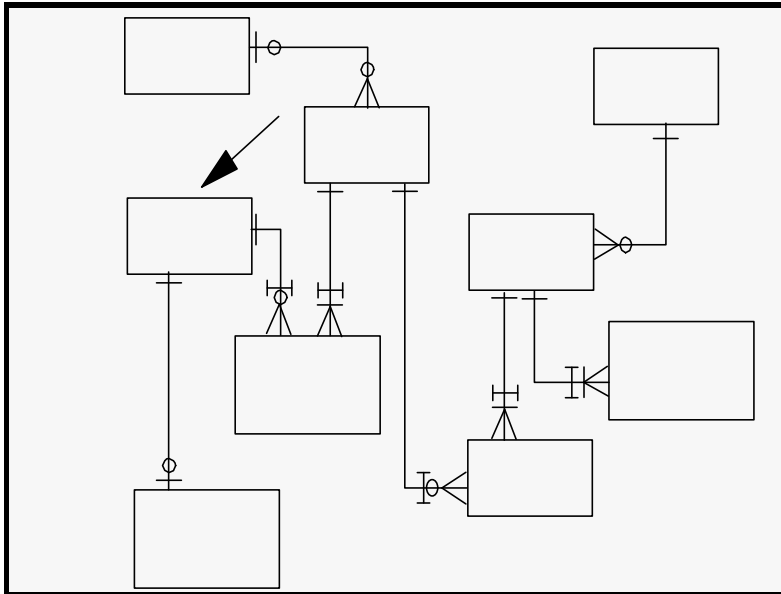
- Missing Delete Rules
  - Explicitly defined DELETE rules, no default usage
- Review for:
  - Missing delete rule specification

Copyright 1996, Nims Associates, Inc.





## *Unacceptable CASCADE/PENDANT Volume*

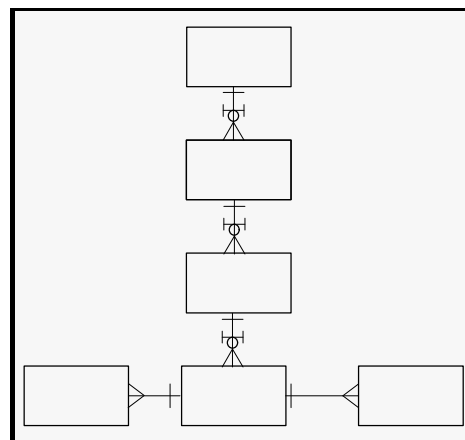


Copyright 1996, Nims Associates, Inc.



## *Long Identifying Hierarchies*

- Can lead to index problems at lower level of hierarchy
- Dependent on concatenated identifier size



Copyright 1996, Nims Associates, Inc.



## **Identifying Hierarchies**

- Long identifier attributes lead to large costly indexes on that entity
- Even costlier indexes on dependents
- Review for:
  - TEXT identifiers
  - Identifying attributes over 9 places wide
  - Combined identifying attributes over 15 places wide
- Suggested Action:
  - Consider alternate identifier (especially if a parent in an identifying hierarchy)

Copyright 1996, Nims Associates, Inc.



## **Key Assignment Documentation**

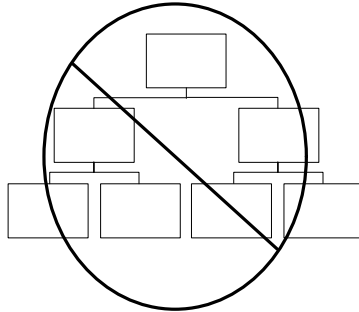
- Each identifier description needs to indicate where the value originates from at CREATE time
- Review for:
  - Identifiers with the property of "design"
  - Identifier names denoting computer assignment
  - Descriptions indicating sequential value assignment
  - Dependent in a non-identifying relationship
- Suggested action:
  - Change assignment to random value

Copyright 1996, Nims Associates, Inc.



## **Key Assignment Documentation**

- Sequential identifier assignments can cause problems
  - Index B-Tree:



Copyright 1996, Nims Associates, Inc.



## **Optional Attributes**

- IEF does not support the concept of NULLs for optional attributes
  - Populated with "default" values if not provided at SET time
  - Beware of default values that also have a real value

<b>Entity:</b> <b>WORK_SCHEDULE_APPT</b>
<b>Attributes:</b> .....
<b>APPT_TIME</b>

Copyright 1996, Nims Associates, Inc.



## **Optional Attributes**

- Review for:
  - Optional attributes
  - Attribute description indicating default value could also appear as a real value
- Suggested action:
  - Define as mandatory attribute
  - Determine substitute for default value
  - Document solution in description of attribute

Copyright 1996, Nims Associates, Inc.



## **Long Character Attributes**

- Problematic if associated row length gets too long
  - Tie with review issue of entity length
- Review for:
  - Attributes over 50 bytes in length
  - 50 is not necessarily too long
  - Multiples of these attributes in one entity type
  - Variance of lengths for same types of attributes across entities
- Suggested action:
  - If long attributes cause for long entities
  - Consider splitting entity into two with 1:1, identifying relationship
  - Consider varying length (see below)
  - Consider compression (CPU costs)

Copyright 1996, Nims Associates, Inc.



## *Varying Length Attributes*

- Implemented as VARCHAR if less than 255 characters
  - Implies processing overhead
  - Useful if attribute contains a true range of lengths, and
  - Attribute is never updated or indexed
  - Must be proven

## *Varying Length Attributes*

- Review for:
  - Property of "varying"
  - Average length
  - Description of use
  - Frequency of update
- Suggested action:
  - If average length is greater than 50% of max or if attribute is updated frequently (generic UPDATES) consider fixed length or separate entity type

## **Incorrect Attributes Properties**

- Attributes need meaningful names, correct properties
  - Attribute names should not conflict with properties
  - Numeric columns are, typically, more efficient than text
- Review for:
  - Attributes named "NUMBER", "NBR", etc. that are text
  - Attributes named "INDICATOR", "IND", etc. larger than 1-byte, text

## **Incorrect Attributes Properties - Date**

- Specific issues occur with END\_DATE attribute
  - For time-dependent data this attribute is usually, logically optional (unknown END\_DATE)
  - If OPTIONAL and not set by AB it will contain default an IEF-supplied of '0001-01-01'
  - Leads to search inefficiencies

Search for entity occurrence with unknown END\_DATE:

```
WHERE DESIRED entity end_date >= CURRENT_DATE  
OR DESIRED entity end_date = datenum(0)
```

## **Poor Attribute Order**

- Identifying attributes should be first in attribute list
  - Less work necessary in technical design
- Review for:
  - Non-identifying attributes before any identifying attributes in attribute list
- Suggested action:
  - Reorder attribute list

Copyright 1996, Nims Associates, Inc.



## **Poor Attribute Order**

- First non-identifying attribute should not be a candidate for an index in technical design
- Review for:
  - Attribute named "NAME" or some derivation as first non-identifying attribute
  - Other attributes that will be used frequently as search criteria in same position

```
REVWIEF: CUSTOMER Attributes
entity CUSTOMER
attr  SSN (identifier)
attr  NAME
attr  HOME_PHONE
```

- Suggested action:
  - Reorder attribute list

Copyright 1996, Nims Associates, Inc.



## **Other Items to For Review**

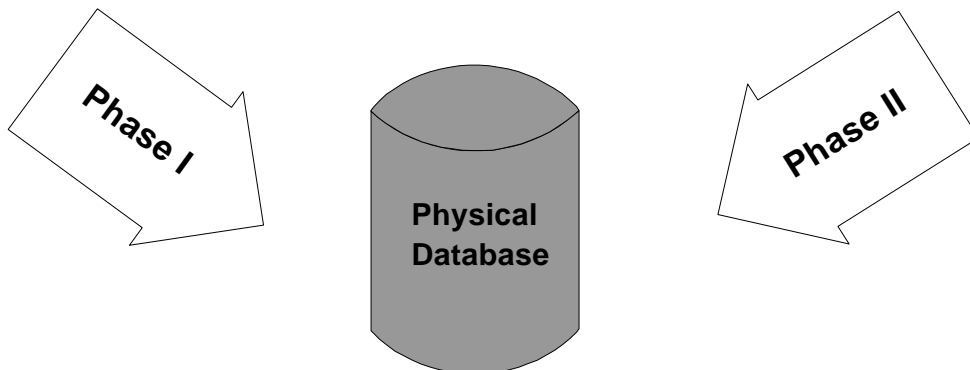
- Isolated Entity Types
- Unimplementable Entity Type and Attribute Names
- Long Permitted Value Lists
- Derived Attributes
- Excessive Entity Type Length
- 1:1 Non-Identifying Relationships

Copyright 1996, Nims Associates, Inc.



## **Part Two: TD Review Objectives**

- Best case scenario: Two Physical Reviews



Copyright 1996, Nims Associates, Inc.





## **Part Two: Phase I TD Review Objectives**

- ✓ *Identify and document 'glaring' physical design issues that are not dependent on processing*
- ✓ *Begin to identify possible processing-dependent issues*
- ✓ *Begin to identify physical design solutions for processing issues*
- ✓ *Document issues to be addressed later*
- ✓ *Insure physical design meets local standards*
- ✓ *Catch problems that may affect ongoing testing*
- ✓ *Proactive design tuning*

Copyright 1996, Nims Associates, Inc.



## **TD Phase I - Review Timing**

- Model transformation timing:
  - Varies shop to shop
  - Our approach:
    - Transform early (and maybe often until model stabilizes)
    - Don't modify TD until data model is very stable
  - Why transform early?
    - Can generate code while building an AB
    - Can test SQL in Query Manager or SPUFI
    - Can find physical problems such as poor row/page fit



Copyright 1996, Nims Associates, Inc.



## **Phase I - Early Data Structure Items**

- Review for:
  - Items that can be addressed with little knowledge of processing requirements
  - Items that may have slipped through ER Technical Review
  - Record/Field names that don't match Entity/Attribute names (unless standard says otherwise)
  - Poor record length/page fit - looking for high unusable space
  - Use of long character columns - cause poor record/page fit
  - Use of VARCHAR columns - are they valid?
  - Placement of VARCHAR columns - are they last?
  - Column likely to be indexed (NAME) as first non-primary key column

Copyright 1996, Nims Associates, Inc.



## **Phase I - Wide Indexes**

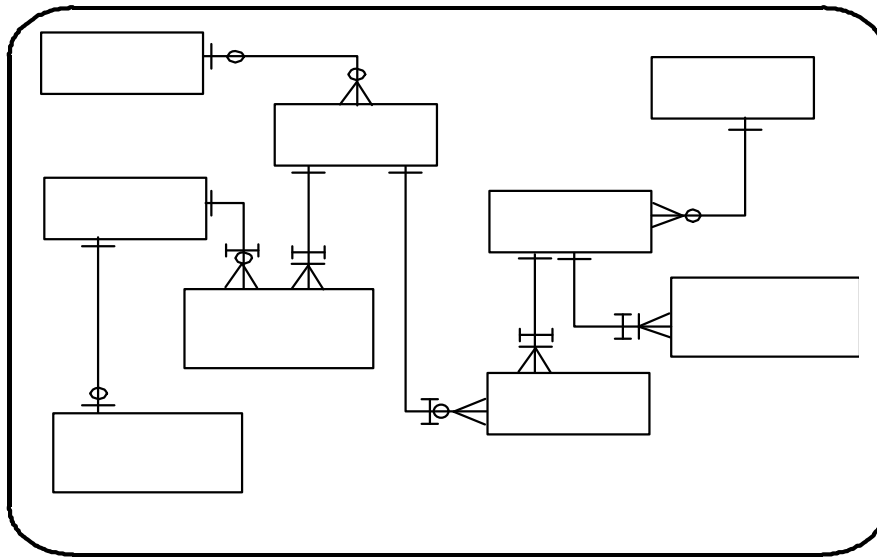
- Indexes/IEF Entry Points
  - Wide indexes leads to high index costs if a lot of data exists
  - Some are unimplementable by IEF
- Review for:
  - Indexes wider than 40 bytes (practical limit, DB2's limit is 254)
  - Indexes with more than 16 columns (IEF implementation limit, DB2's is 64)
- Suggested action:
  - Consider worth of index if it is not the primary key index
  - Consider re-identifying the entity or breaking a long identifying hierarchy in the model if the problem is the primary key index

Copyright 1996, Nims Associates, Inc.



## **Phase I - RI Support**

- With DB2 RI chosen



Copyright 1996, Nims Associates, Inc.



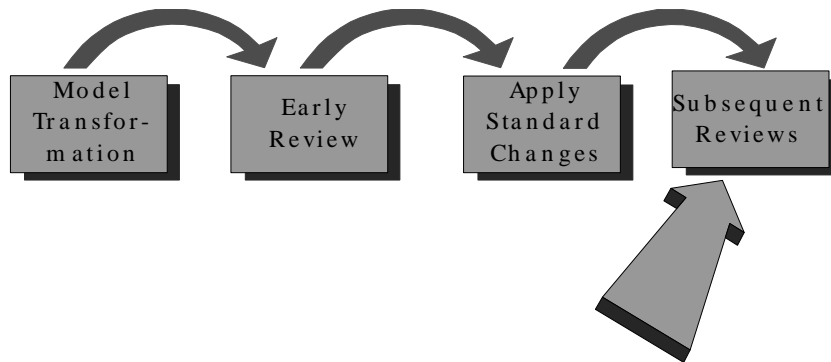
## **Phase I - Early Data Store Items**

- Typically come from defaults
  - Shop standards may vary
- Review for: items that don't meet standards or suggestions below
  - Database, Tablespace and Indexspace bufferpool usage - See shop standards
  - STOGROUP vs. VCAT - See shop standards
  - STOGROUP/VCAT names - See shop standards
  - Index Subpages - Suggest "1"
  - LOCKSIZE - Suggest "PAGE"
  - CLOSE - Suggest "NO"
- Suggested action:
  - Change defaults to meet standards/suggestions

Copyright 1996, Nims Associates, Inc.



## **Phase I - Subsequent Data Structure Items**



Copyright 1996, Nims Associates, Inc.



## **Phase I - Subsequent Data Structure Items**

- Review for these after:
  - Standard changes are made
    - Automation of review items can be used to create input for standard changes
  - TD has been constructed as tables
    - Enables comparison of IEF to DB2
- Delay changes until ER model is stable
  - Changes are lost if ER object is reimplemented in TD

Copyright 1996, Nims Associates, Inc.



## **Phase I - Standard Object Names**

- Shop standards may cause need to change default object names
  - Re-implementation of ER object resets names back to default
- Review for:
  - Record/Table, Entry Point/Index, Linkage/Foreign Key Constraint names that don't meet standards
- Suggested action:
  - Update TD

## **Phase I - Standard Object Names**

- Review automation example:

```
SELECT SUBSTR(R. MACRO_NAME, 1, 18),  
       SUBSTR(F. MACRO_NAME, 1, 18), E. NAME, A. NAME  
FROM HIEFDD53. RECORD R,  
     HIEFDD53. FIELD F,  
     HIEFDD53. ATTRIBUTE A,  
     HIEFDD53. ENTITY_TYPE E  
WHERE F. MODEL_ID = R. MODEL_ID  
      AND F. RECORD_IS_IN_ID = R. ID  
      AND F. MODEL_ID = A. MODEL_ID  
      AND F. ATTRIBUTE_IS_IN_ID = A. ID  
      AND A. MODEL_ID = E. MODEL_ID  
      AND A. PARENT_ENTITY_ID = E. ID  
      AND F. MODEL_ID = ???  
      AND F. MACRO_NAME LIKE ' FK_%'
```

## *Phase I - Nullability*

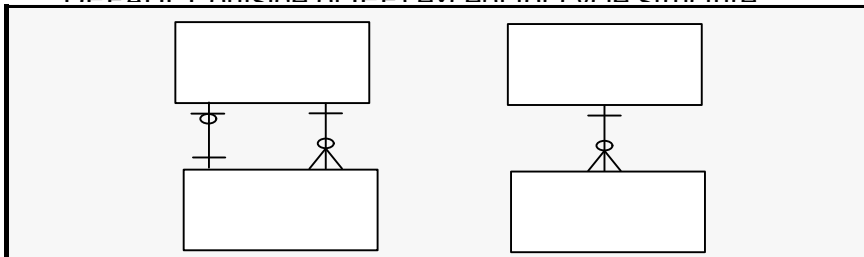
- IEF creates many field/column definitions as NULLable
  - Most will never contain NULLs
  - Optional and Subtype attributes
    - Concept of NULLs may be supported in future releases
    - NULLable definition adds storage requirements and CPU overhead
  - Foreign keys in mandatory-parent relationship
    - Create a possible RI hole
    - Necessary only for support of mandatory parents in a complete cycle

Copyright 1996, Nims Associates, Inc.



## *Phase I - Nullability*

- Review for:
  - Non-foreign key fields defined as NULLable
  - Foreign key fields defined as NULLable
- Suggested action:
  - Change non-foreign key fields to NOT NULL (or NOT NULL WITH DEFAULT outside of IEF)
  - Change foreign key fields to NOT NULL (or NOT NULL WITH DEFAULT outside of IEF) except for cycle structure



Copyright 1996, Nims Associates, Inc.



## **Phase I - Primary Key Placement**

- Column placement is based upon attribute order and relationship membership
  - Dependents in identifying relationships will always have part of the primary key at the end of the row by default
  - Easier to develop with, administer, and perform ad-hoc queries if primary key columns are at beginning of row
- Review for:
  - Existence of a non-PK column before the last PK column

## **Phase I - Primary Key Placement**

- Suggested action:
  - Rearrange field/column order

```
Record      +-EMERGENCY_CONTACT      76
Field      SEQ_NUMBER      Small      2      Not Null
Field      NAME      Text      35      Not Null
Field      RELATIONSHIP      Text      16      Null
Field      PHONE      Packed      6      Not Null
Field      EXTENSION      Integer      5      Null
FK Field    FK_ENTRANTSSN      Integer      4      Not Null
Linkage     <No Name> ENTRANT
EntryPoint (U) +-I0000413      Primary      6
Field      FK_ENTRANTSSN      Integer      4      Not Null
Field      SEQ_NUMBER      Small      2      Not Null
|      +-
|      +-
+-
```

## **Phase I - Additional Indexes**

- IEF builds indexes on primary keys and most foreign keys
  - Additional indexes will probably be added as applications are built
  - IEF generates FOR UPDATE OF clause on parent in ASSOCIATE, DISASSOCIATE, and TRANSFER based upon Record layout
  - Don't want FOR UPDATE OF targeting indexed columns
- Review for:
  - Records/Tables whose first non-identifying column is indexed
- Suggested action:
  - Optimal - Change attribute order in ER model and reimplement object

Copyright 1996, Nims Associates, Inc.

OR Change order of fields/columns in TD



## **Phase I - Additional Indexes**

- Example of first nonidentifying column indexed

Record	+- ENTRANT		79	
Field	SSN	Integer	4	Not Null
Field	NAME	Text	25	Not Null
Field	BIRTH_DATE	Date	4	Not Null
Field	SEX	Text	1	Not Null
Field	ADDRESS	Text	35	Not Null
FK Field	FK_BRACKETCODE	Text	2	Null
Linkage	<No Name> BRACKET			
EntryPoint	+- ENTINX2		25	
Field	NAME	Text	25	Not Null
	+-			
EntryPoint	+- I0000431		2	
Field	FK_BRACKETCODE	Text	2	Null
	+-			
EntryPoint (U)	+- I0000373 (Primary)		4	
Field	SSN	Integer	4	Not Null
	+-			

Copyright 1996, Nims Associates, Inc.





## **Phase I - RI Rule Implementation**

- Physical RI definitions are stored in "Linkages"
  - Some "bugs" exist in certain toolset releases
    - ER rule and TD rule can get out of sync
    - "RI Process" can get reset to default enforcement
  - Occurs when objects are "reimplemented" in TD
- Review for:
  - Linkage RI rules that don't match ERD relationship RI rules
- Suggested action:
  - Run RI Process

Copyright 1996, Nims Associates, Inc.



## **Phase I - Space Allocation**

- IEF defaults freespace to PCTFREE=10 and FREEPAGE=0
  - May be too much or not enough
  - Could mean there is no freespace
  - Can indicate that growth has not been considered
- Review for:
  - Settings of PCTFREE=10
  - Unusable percent > PCTFREE (no freespace)
- Suggested action:
  - Determine proper freespace and set accordingly

Copyright 1996, Nims Associates, Inc.



## **Phase I - Space Allocation**

**Unusable space percent =**

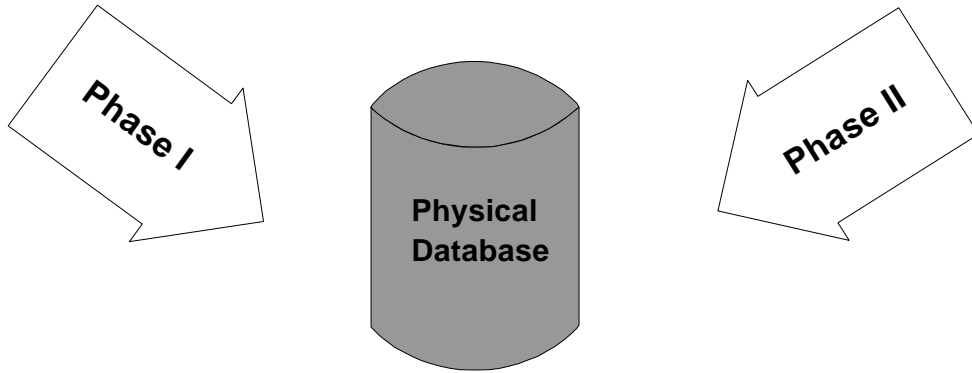
$$100 * ((4074.0000 / \text{RECLength} - 4074 / \text{RECLength}) * \text{RECLength}) / 4074.0$$

## **Phase I - Other TD Review Items**

- RI Across Subject Areas
- IEF vs. DB2 Definition Mismatch
- Hidden RI Due To IEF Delete Processing
- Add Clustering Indexes
- Add DB2 Table Partitioning

## **Part Two: TD Review Objectives**

- Best case scenario: Two Physical Reviews



Copyright 1996, Nims Associates, Inc.



## **Part Two: PHASE II TD Review Objectives**

- ✓ Realize the technical impact of IEF's install logic
- ✓ Review the DB2 EXPLAIN output to identify problematic access
- ✓ Define the need to review index column order
- ✓ Identify unnecessary indexes built by IEF
- ✓ Describe what to review for to prevent DELETES from causing multiple tablespace scans
- ✓ Proactive design tuning - determine how the DBMS will handle the applications in the target environment
  - ▶ Identify access path issues
  - ▶ Identify unnecessary objects
  - ▶ Identify plan definition issues
- ✓ Review applies the production environment statistics to the physical design for the processing defined

Copyright 1996, Nims Associates, Inc.



## **TD Phase II - Review Timing**

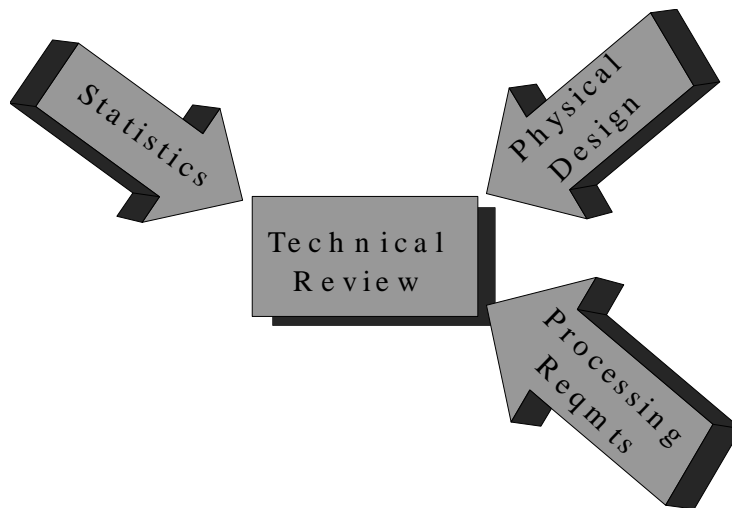
- To be held after the bulk of action diagrams are developed, constructed and installed
- Broken into three categories - not necessarily separate
  - EXPLAIN analysis
    - Examination of how the code will access the data
  - INDEX analysis
    - Examination of how useful the indexes are
  - PLAN analysis
    - Final tuning of plans
- Consists of working sessions and formal review - similar to ERD Reviews



Copyright 1996, Nims Associates, Inc.




## **Phase I I - What are the Review Materials?**



Copyright 1996, Nims Associates, Inc.



## **Phase I I - Statistical Information**

- Target environment statistical information is critical to success of this review
  - All statistics available in or can be extrapolated from IEF statistics **except** column cardinality
  - Very crucial for proper index design and for index usage decisions
- **Column Cardinality**  Number of distinct values in a column
- Gather information from end-users
  - OR ---
- Copy from current system production statistics

Copyright 1996, Nims Associates, Inc.



## **Phase I I - EXPLAIN Analysis**

- All applications EXPLAINed with production statistics
  - Report on problematic access
  - Review code causing problematic access
  - Document problematic access solution point, classify as
    - Appl: Application code issue
    - DBA: Physical data design issue
- Typically, index needs
- Document desired change and expected benefit by table

Copyright 1996, Nims Associates, Inc.



## **Phase I I - EXPLAIN Analysis**

- After examining all problematic access, physical design requirements will end up grouped by table, for example

**Table:** CUSTOMER

**Column:** NAME

Action Diagram: XXX\_LIST\_CUSTOMERS - Qualifying to customer and sorting on NAME. Heavily used PrAD. Suggest index, ascending order.

## **Phase I I - EXPLAIN Analysis**

- Items discovered during EXPLAIN analysis include:
  - DB2 sorts executed & why
  - How joins are processed by DB2
  - Lock strategy chosen by DB2
  - Prefetch indications
  - Index usage
    - Multiple index usage
    - Which index is being used
    - Which index columns are matched
  - Tablespace scans

## **Phase I I - Post-EXPLAIN Analysis**

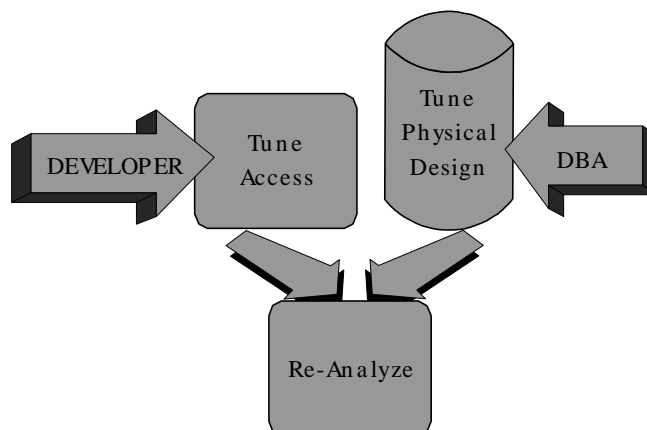
- Two deliverables:
- Applications that need tuning
  - Developers tune applications based upon action diagram guidelines and weighting factors
- Physical design requirements by table
  - DBA's implement physical design requirements if:
    - Benefits outweigh costs - Helpful to have a report listing access requirements by table (READ requirements by ENTITY)
    - Design is static - Development is fairly complete

Copyright 1996, Nims Associates, Inc.



## **Phase I I - Post-EXPLAIN Analysis**

- Re-run analysis to verify impact



Copyright 1996, Nims Associates, Inc.



## **Phase I I - Index Analysis**

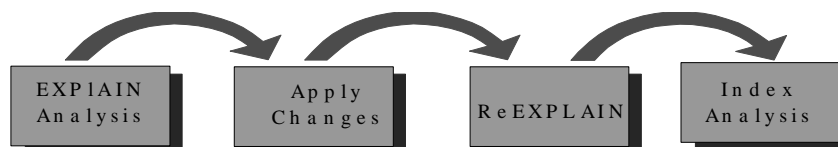
- Follows completion of EXPLAIN analysis
  - Assumes changes recommended in EXPLAIN analysis are implemented
  - Assumes applications are reEXPLAINed following changes
- Goal - determine if data design contains valid index definitions
- Utilizes EXPLAIN output
- Helpful to have:
  - A report listing ORDER BY requirements by table (READ EACH/SORTED BY requirements by ENTITY)
  - Summary-level usage reporting of access types by index

Copyright 1996, Nims Associates, Inc.



## **Phase I I - Index Analysis**

- Use repetitive analysis, make sure what was changed for one performance reason did not 'break' something else



Copyright 1996, Nims Associates, Inc.





## **Phase II - Index Analysis Deliverables**

### ■ **Scanned indexes**

- Indexes that are always or usually accessed via a scan
- Usually, indicates poor column ordering within the index
- Suggested action:
  - Consider reordering the index column order based upon application usage requirements
  - Ensure that changing the index column order will not cause more physical sorting

Copyright 1996, Nims Associates, Inc.



## **Phase II - Index Analysis Deliverables**

### ■ **Unused Index columns**

- Some multi-column indexes may have columns never being used
- Suggested action:
  - Determine if a different column order would cause all column usage
  - If no other order is better and if index is not being used for uniqueness or RI purposes (see RI point below) consider removing unused columns from index

Copyright 1996, Nims Associates, Inc.



## ***Phase II - Index Analysis Deliverables***

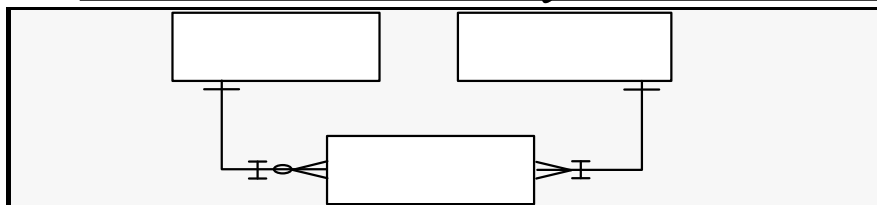
### ▪ **Duplicate indexes**

- A duplicate index is an index whose columns and column order fully or partially match the high-order columns of another index on the same table
- Wasted indexes cause additional space and CPU overhead
- IEF creates many of these on "associative" tables
- Suggested action:
  - Remove index overhead
- NOTE: Also, look for indexes that are duplicates if column order changes
  - Could hurt access paths

Copyright 1996, Nims Associates, Inc.



## ***Phase II - Index Analysis Deliverables***



Record	+--DIRECTOR_ASSIGN		39	
Field	POSITION	Text	25	Not Null
FK Field	FK_EVENTNUMBER	Small	2	Not Null
FK Field	FK_DIRECTORSSN	Integer	4	Not Null
Linkage	<No Name> DIRECTOR			
Linkage	<No Name> EVENT			
EntryPoint (U)	+--I0000399 (Primary)		6	
Field	FK_DIRECTORSSN	Integer	4	Not Null
Field	FK_EVENTNUMBER	Small	2	Not Null
	+--			
EntryPoint	+--I0000396		4	
Field	FK_DIRECTORSSN	Integer	4	Not Null
	+--			
EntryPoint	+--I0000393		2	
Field	FK_EVENTNUMBER	Small	2	Not Null
	+--			
	+--			

Copyright 199

## **Phase II - Index Analysis Deliverables**

- **Unused indexes**
- Some indexes generated are never used by any defined access
  - These indexes are candidates for removal
  - Consider RI issues - don't remove indexes needed for DB2 RI (see following item)
- Suggested action:
  - Consider removing unused index if there is no RI dependency

## **Phase II - Plan Analysis**

- Final, generic tuning of plans
- Typically few problems found
- Important to review for in stress and high production volume environments

## *Phase II - Plan Analysis*

- What to review for and suggested actions:
  - Plans marked VALIDATE='R'
    - Rebind plans with VALIDATE(BIND) option
  - Plans bound as Repeatable Read
    - Rebind plans with ISOLATION LEVEL(CS)
  - Plans with CACHESIZE > 0 and plan authority granted to public
    - Rebind plans with CACHESIZE(0)
  - Tables not used
    - Determine if processing is missing

Copyright 1996, Nims Associates, Inc.



# Questions?? Comments??

■ ???????

Copyright 1996, Nims Associates, Inc.

